AD-A155 753

1

THE ARPANET IMP PORT EXPANDER

Technical Report 1080-140-1

November 1980

By: Holly A. Nelson, Systems Programmer
James E. Mathis, Research Engineer
James M. Lieb, Research Engineer

Telecommunications Sciences Center
Computer Science and Technology Division

This research was sponsored by the Defense Advanced Research Projects Agency under ARPA Order No. 2302, Contract MDA903-80-C-0222, monitored by Dr. B. Leiner.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the United States Government.

SRI Project 1080

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION IS UNLIMITED (A)

DTIC
ELECTE
JUN 1 8 1985
S D
G

SRI International
333 Ravenswood Avenue
Menlo Park, California 94025
(415) 326-6200
Cable: SRI INTL MPK
TWX: 910-373-1246

DTIC FILE COPY

SRI International

85 06 13 026

CONTENTS

APPENDICES

Accession For

NTIS GRA&I ☒
DTIC TAB □
Unannounced □
Justification______

By______
Distribution/
Availability Codes

Dist	Avail and/or Special
A-1	

DTIC
COPY
INSPECTED
3

ILLUSTRATIONS

TABLES

I INTRODUCTION

The ARPANET was originally conceived to support high-data-rate distant communication between large mainframe computers. Because of the hardware limitations of the Honeywell 316/516 processor, which was selected as the original ARPANET interface message processor (IMP), most ARPANET IMPs are restricted to supporting a maximum of four host attachments (computers or gateways to other networks).

As the ARPANET community expanded many sites experienced the need for more than four host ports. The response was to install additional IMPs. With the H316/516 no longer available and the C/30 IMP, being developed by the BBN Computer Company not expected to arrive until 1981, no more ARPANET nodes/host ports were available. Simultaneously with this growth in the ARPANET community, hosts attached to IMPs have increasingly tended toward less powerful types, such as the DEC PDP-11 minicomputer. Many of these minicomputer hosts require only moderate data-rate network communication or occasional access.

Newer node-switching equipment utilizes the Bolt Beranek and Newman (BBN) Pluribus, which eliminates hardware constraints upon the number of host attachments. Nevertheless, the large majority of deployed IMPs are still of the older Honeywell type. The desire to attach a large number of minicomputers or microcomputers to the ARPANET induced DARPA to assign SRI the task of developing the port expander concept into a working product. This manual describes the functions, installation, and operation of the port expander.

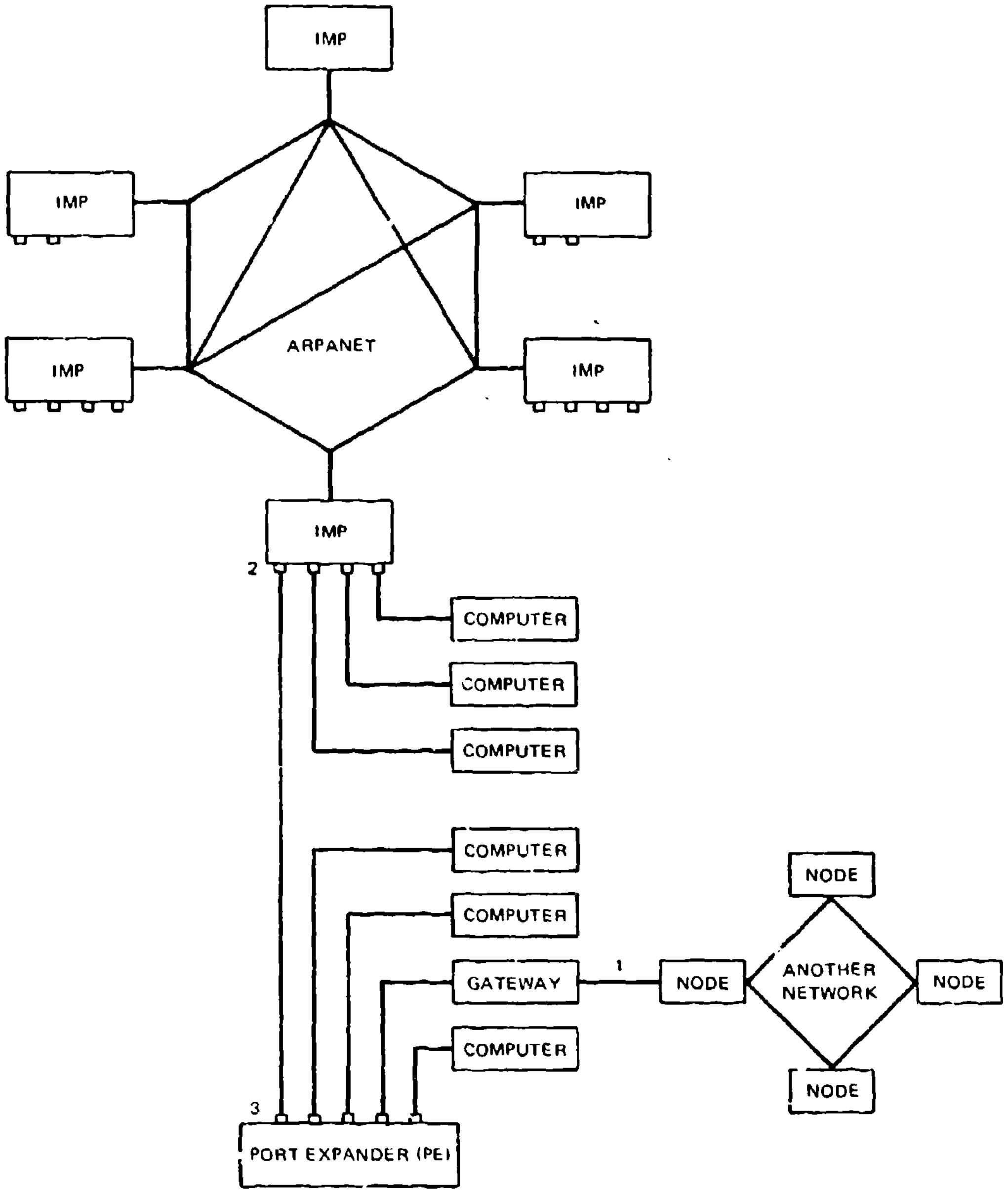
II THE PORT EXPANDER CONCEPT

The port expander (PE) permits up to four hosts to share one IMP host port, as shown in Figure 1. Once a host is interfaced with the ARPANET, substitution of a connection by means of the port expander in place of the host's direct IMP attachment is transparent to hardware and software in both the IMP and the host. The port expander acts as a host to the IMP, as an IMP to its hosts, and entails no changes to the ARPANET subnetwork.

A. Destination-Address Demultiplexing

In any demultiplexing scheme, some set of attributes must be used as a basis for deciding which of the several hosts are to receive a message. Usually the demultiplexing is based on destination-address information conveyed in the message leader or header. Unfortunately, the ARPANET IMP has a rather myopic view of the "outside" world; the port expander and its attached hosts are all perceived as a single host by the ARPANET. This inability of the network to distinguish among the hosts on the port expander forces the demultiplexing to be based on something other than the ARPANET address. Since the Network Control Protocol (NCP) employs the link and subtype fields of the ARPANET message leader in a manner that is not suitable for demultiplexing, some additional information not contained in the ARPANET leader must be supplied.

This demultiplexing information is supplied to the port expander by requiring all but one of the attached hosts to use the internet protocol, rather than the ARPANET-specific NCP. The port expander examines the destination address field in the internet header, when present, to determine which of the attached hosts should receive the message; if no internet header is present, the message goes to a default NCP host.



NOTES:

1. Any host port can be used as a gateway port to another network. Only one NCP host can be attach^ed to a PE.
2. Any IMP host port can have a PE attached.
3. All PE interfaces are 1822 distant host type.

FIGURE 1 TH: PORT EXPANDER ENVIRONMENT

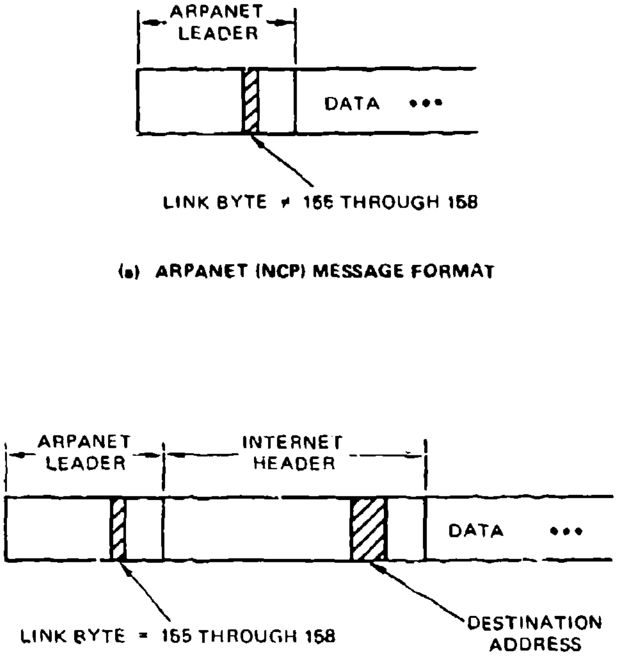
Internet protocol traffic is distinguished from NCP traffic by the link number in the ARPANET leader. The internet links are Nos. 155 through 158 (decimal), and packets on these links must have an internet header following the ARPANET leader. The NCP traffic, on the other hand, is used on link numbers other than 155 through 158; these packets do not have an internet header after the ARPANET leader. Figure 2 illustrates this use of link numbers.

The PE can be programmed to direct all NCP traffic, which does not have internet headers, to one of the attached hosts. This allows one NCP-based host and several TCP-based hosts to be serviced by the PE off the same IMP port. This is an important restriction: the port expander cannot be used to multiplex several NCP-based hosts onto one IMP port.

B. Packet Routing

In routing packets from an attached host, the port expander forwards everything to the IMP except packets between attached hosts and packets intended for the PE itself. In routing packets from the IMP, the port expander's NCP host (if present and enabled) gets all packets that do not use internet protocol; internet packets, however, are routed according to a programmable "routing table," which can be changed at the console by the operator.

The state of the optional NCP host, which can be attached to any single port on the PE, will affect the routing of internet messages. If the NCP host goes down, no packets can be sent out to the port expander's IMP.



(a) ARPANET (NCP) MESSAGE FORMAT

(b) INTERNET MESSAGE FORMAT

FIGURE 2 PORT EXPANDER MESSAGE FORMATS

C. Major Features

In summary, the salient functional capabilitie and restrictions of the port expander are as follows:

- It supports at most one NCP-based host and up to four internet-based hosts.
- It is compatible only with the new 96-bit ARPANET leader.
- Its operation is functionally transparent to the hosts (i.e., the port expander looks exactly like an IMP to the hosts).
- The additional store-and-forward delay is less than or equivalent to one IMP-to-IMP hop.
- If an NCP host is used, all message routing stops when the NCP host goes down.

Hardware specifications for the port expander are given in the next section.

III HARDWARE PHOTOGRAPHS AND SPECIFICATIONS

The photographs in Figures 3 and 4 provide front and back views of the two types of port expander hardware. These types are

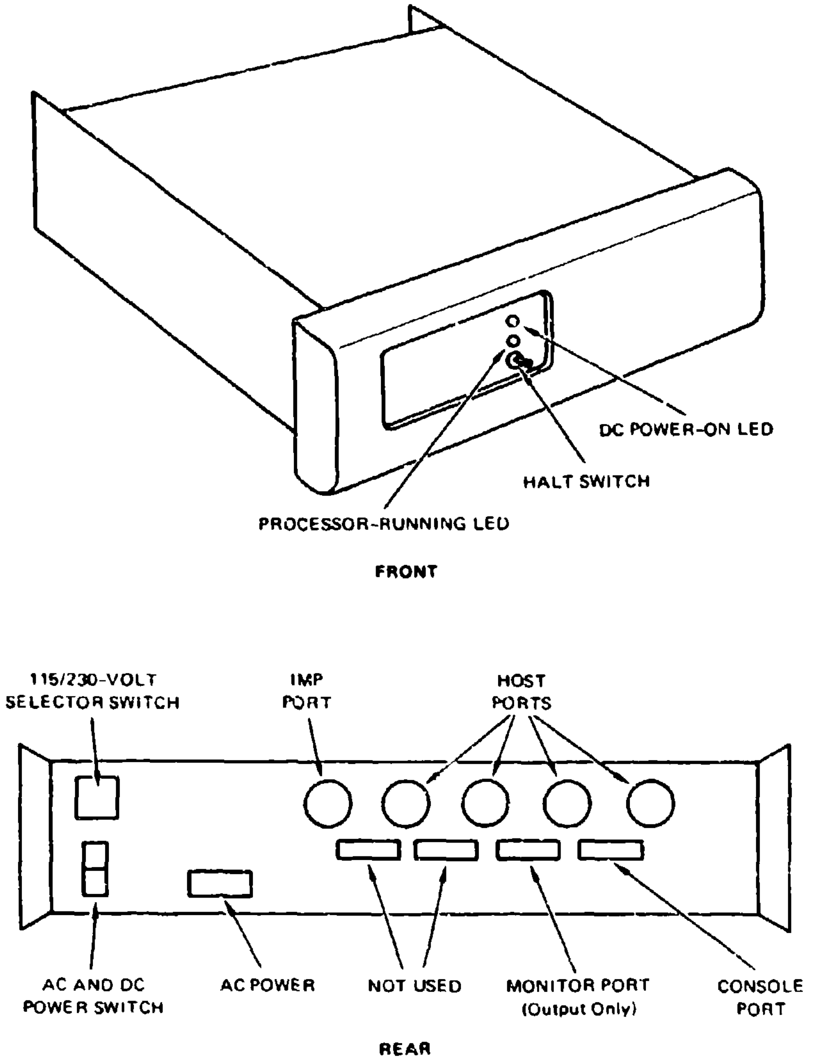
. Standard DEC configuration

. Ruggedized configuration.

Only the standard DEC configuration is currently available. The ruggedized configuration is an early version of the port expander used in applications subject to shock and vibration. The normal DEC configuration is sui'able for use in commercial computer-type environments.

Figure 3, the standard DEC configuration, shows two LEDs and one switch on the front panel. The top LED indicates when both the +5-volt and the +12-volt dc power are on. The LED below this indicates when the central processor is running. When depressed, the spring-loaded switch below the LEDs causes the processor to halt.

The rear view in Figure 3 shows the power switch for both ac and dc on the lower left. Above it is a strapped selector switch for selecting ac voltage (115 or 230). Along the top are the cable connectors for the IMP and the hosts; this photograph shows one IMP connector (extreme left) and four host connectors. Along the bottom are four RS-232 connectors; the rightmost connector is for the operator's console and the one next to that, marked "2," is for an output-only trace and message monitor. The remaining RS-232 connectors are not used. All RS-232 ports are hardwired to operate at 9600 baud, although other common speeds can be selected by changing the hardware jumpers commensurately.



FRONT

REAR

FIGURE 3 STANDARD DEC CONFIGURATION

Figure 4, the ruggedized configuration, shows a somewhat different arrangement on the front panel. Tne red ac-power switch and light are on the left side. Along the top right are two LEDs and three switches that are normally covered with a transparent panel, except during maintenance operations: the left LED indicates the presence of +5- and +12-volt dc power and the right one indicates that the central processor is running; the three switches, which are all normally in the up position, include (1) a dc power switch, (2) an enable/halt switch for the CPU, and (3) a line time clock (LTC) switch in which the up position enables a 60-Hz clock. In some cases, there are four baud rate switches mounted on the lower righthand side of the front panel. Each corresponds to one of the RS-232 ports at the rear, expect that the one labeled "3" matches the port marked "console." The nameplate on the front panel is removable to provide access to the internal components for maintenance.

The rear view in Figure 4 shows the host cable connectors on the left side; up to four can be installed. The IMP cable connector, labeled "1822," is at the upper right. Only two of the four RS-232 connectors are used; the top one (marked "console") is for the operator's terminal and the second one down (marked "TTY 2") is for the output-only trace monitor.

Table 1 gives the key hardware specifications for the port expander. More information on the types of boards included in the system are given in Section IV, along with references to documents that describe the boards in detail.

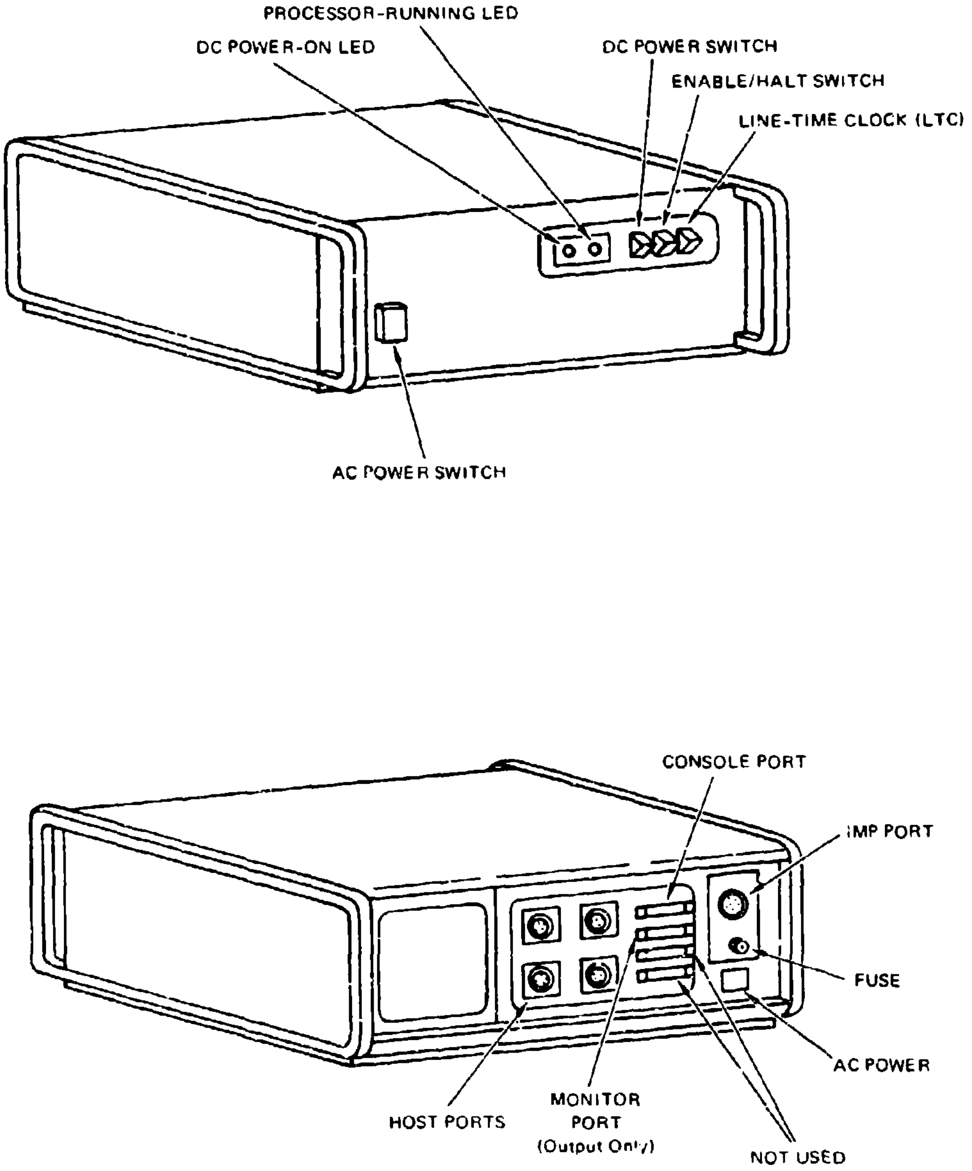


FIGURE 4 RUGGEDIZED CONFIGURATION

Table 1

PORT EXPANDER HARDWARE SPECIFICATION

CPU: LSI-11/2

MEMORY: 28K WORDS

PORTS:	HOSTS[2]	IMP	CONSOLE	MONITOR
• NUMBER (min)	1	1	1	1
• NUMBER (max)				
STANDARD DEC	4	1	1	1
RUGGEDIZED	4	1	1	1
• DIRECTIONALITY	I/O	I/O	I/O	O
• INTERFACE TYPE	1822[2] DISTANT HOST	1822 DISTANT HOST	RS-232C	RS-232C
• SPEED (baud)				
STANDARD	na	na	9600[1]	9600[1]
MINIMUM	na	na	150	150
MAXIMUM	180K	180K	38.4K	38.4K

NOTES: 1. Ports are wire wrapped at standard speed unless specificed otherwise.

2. Port-expander compatibility has been proven with hosts using the following interfaces:

DEC IMP-11-A
ACC LH/DH-11
DEC AN-10

IV INSTALLATION AND STARTUP

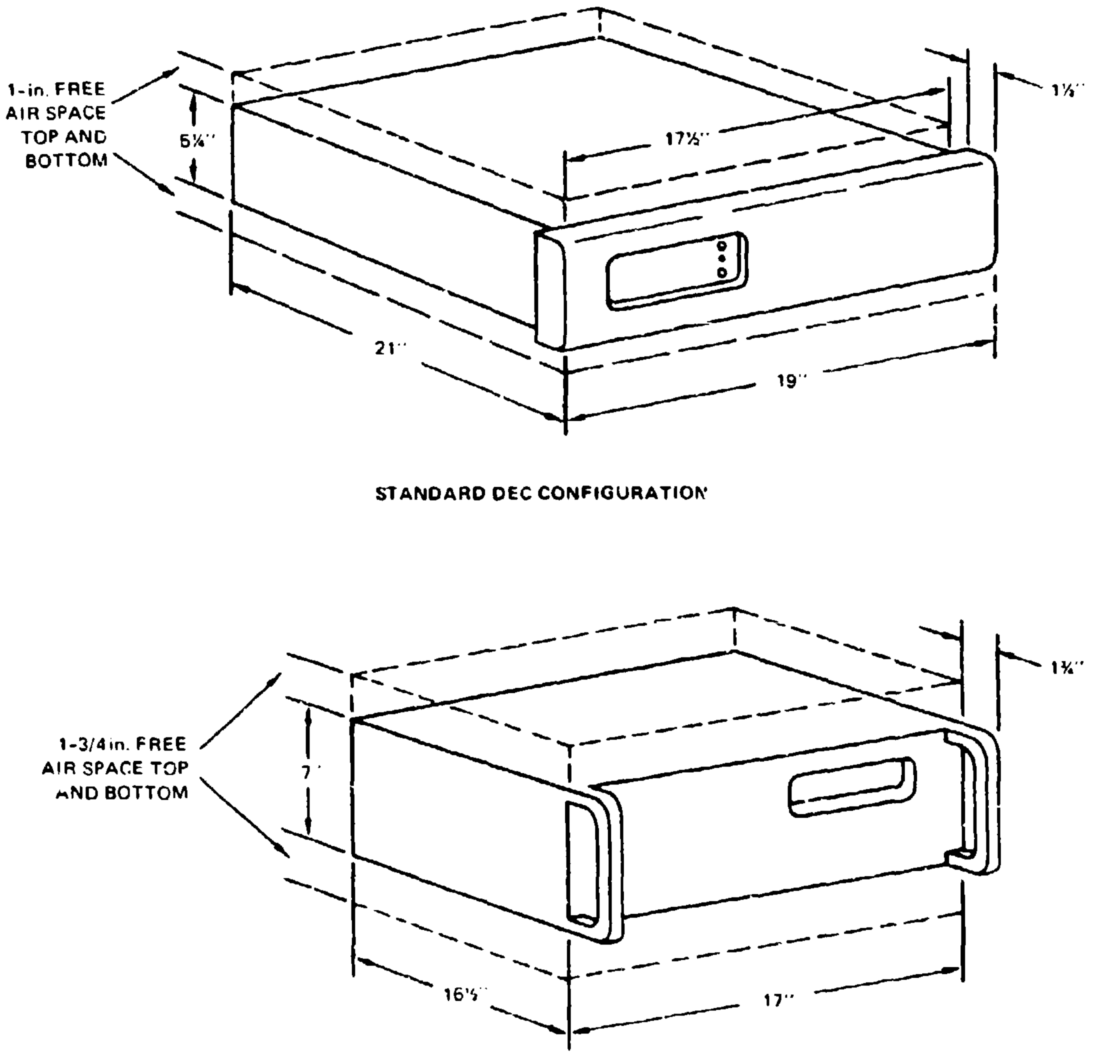
The port expander is shipped as a fully assembled unit, with a power cord. By special arrangement, SRI will provide a 100 foot cable to connect the port expander to the IMP. On-site installation will be provided by SRI after users have prepared the mounting space and cables described below.

A. Power and Environmental Requirements

The port expander will operate from a primary power source of 60-Hz, 115 Vac, single-phase, at 200W maximum. It is rated for operation over an ambient temperature range of 5^{o}-40^{o} C, with a noncondensing relative humidity of 10% to 95%. Of course, for maximally reliable operation, an environment suitable for most digital computer equipment should be provided; i.e., air-conditioned and free from excessive dust, moisture, or corrosive fumes.

B. Mounting Space

The hardware is contained in a deep instrument case designed for mounting in a 19" rack, as shown in Figure 5. A free-air clearance above and below the case is recommended for ventilation, as well as a rear clearance for easy connector access. These clearances are indicated in Figure 5


1-in. FREE AIR SPACE TOP AND BOTTOM
5¼"
17½"
1½"
21"
19"

STANDARD DEC CONFIGURATION

1-3/4in. FREE AIR SPACE TOP AND BOTTOM
7"
1¾"
16½"
17"

RUGGEDIZED CONFIGURATION

FIGURE 5 INSTALLATION SPACE REQUIREMENTS

C. Cable Connectors

The following connectors and cables are supplied:

- A primary-power cable 6 1/2 ft. long, terminating in a molded three-pronged plug.
- Four serial EIA RS-232 terminal interface connectors (DB-25), wired according to the RS-232-C specification as data communication equipment (DCE) (as indicated in Appendix B). See Figures 3 and 4 regarding use of each connector.
- Up to five 1822 distant-host interface bulkhead connectors of the standard type (Amphenol MS24264R18831SN). One connector is for the IMP, and the remainder for hosts. See Figures 3 and 4.

Users are required to supply the necessary cables for attaching the port expander to the IMP and connecting the hosts. Upon request, SRI will furnish a 100-foot cable to connect the port expander to the IMP. Appendices B and C list pinouts for the cable connectors. Note that the distant-host type of interface is provided for both the IMP and host ports. The port expander has operated successfully with hosts using the following standard types of 1822 distant-host interfaces:

- DEC IMP11-A
- DEC AN-10
- ACC LH/DH-11.

D. Board List

Since on-site assistance will be provided by SRI for the initial installation, this equipment checklist is intended for reference only. The port expander card cage contains the following boards:

	Board	ID	Manual
(1)	Line-time clock and power sequencer board (included in ruggedized configuration only)	M8016Y	DEC
(2)	LSI-11 processor board with the KEV-11 extended instruction option	M7270	DEC

(3)	32K-word RAM memory board (MSV11-D configuration)	M8044	DEC
(4)	Direct-memory-access controller board	MICRO-11	ACC
(5)	Up to 5 1822 distant-host interface boards	1822	ACC
(6)	One 4-port serial EIA RS-232C interface board	M8043	DEC
(7)	SRI robustness board with watchdog timer	none	TIU

The references in "Manual" column, are detailed below.

E. Documentation Checklist

Prior to actual delivery and installation of the port expander, the user's technical staff should gain familiarity with its hardware and operational procedures. The following documentation is included:

. TIU Notebook (Volumes 1 through 4)

. DEC Microcomputer Processor Handbook

. DEC Memory and Peripherals Handbook

. ACC Micro-11 Maintenance Manual.

F. Software Downloading and Startup

Software is either cross-net-loaded from a network loader-server or downloaded from a local host. Cross-net loading takes place through the port expander's IMP 1822 port and is the normal, unattended mode of operation. Downloading takes place through the RS-232 port reserved for the console terminal; the console is temporarily disconnected, so that its RS-232 port can be connected to the local host during this operation. When cross-net loading is used, loading is done on power-up or recovery from a power failure using the robustness card. The robustness software/hardware is documented in the TIU notebook, Vol. 1 Section III, and Vol. 2 Section IV.

The exact manner of downloading and startup will be established during installation by an SRI representative.

For debugging purposes, the downloaded binary (.BIN) file contains the DDT debugger and on start-up the following herald is printed out on the console

```
DDT, <address>
```

After this appears, type a carriage return and then a $START<esc>G, as follows:

```
<cr>
$START<esc>G
```

This will cause the monitor terminal to display

```
SRI port expander <version>, <month>.<year>
```

It will also cause the console terminal to display

```
PE console, Version <version>
```

For normal operation, the .BIN file does not contain DDT and the port expander will load and start itself when the power switch is turned on.

Normal startup messages seen on the monitor terminal after this point are as follows:

Message	Meaning
'Port <n> - Error on Port'	This message always occurs on restart once for each attached host. It indicates a ready-line transition.
'NOP from IMP Host/IMP: <oct>/<oct> Link: <oct> Type: <dec> Port: <oct> Op: <oct>'	This contains information from the ARPANET leader received from the attached IMP, the port number that the packet was sent from, and the signal opcode value. See "Monitor Messages" in Section V.
'NOP from Host Host/Imp: <oct>/<oct> Link: <oct> Type: <dec> Port: <oct> Op: <oct>'	This is the same information as in the above message for a NOP packet from an attached and active host (i.e., a host that sends a NOP upon startup or when an error occurs).

The messages are not necessarily in the above order. Other so called "error" messages are also normal during startup. For example:

Message	Meaning
'Received old message format'	This indicates that the IMP has not decided what kind of leader the PE requires (32 or 96 bit).
'Error in leader'	This occurs when the communication with the IMP is reset.

If any of these messages continue, there is a problem. Otherwise the port expander is up. After startup, the only message that will be seen is an occasional "No route for message Host/IMP..." when a stray packet wanders in. For a complete list of possible messages and more details on their meaning, see the end of Section V.

If software is downloaded from a local host, the cable that was the download source must be removed and the console terminal's cable replaced in the port-expander's console socket upon completion of downloading. Tne port expander must not be left attached to the download source after downloading is completed, except when Break is disabled on the DLV11-J terminal-interface board.

Route tables for the port expander are location-specific. Based on information provided by the user, a route table matching your location will be generated and included in the binary (.BIN) file downloaded into your port expander.

V BASIC OPERATION

After the port expander is powered up and loaded with software, it is normally left to run indefinitely. Operator control is needed only wnen ports are to be reconfigured or when error conditions are encountered.

The port expander performs its functions in a manner that is generally invisible to the operator. Normal packets are received, routed, and sent without any external evidence thereof at the console or monitor terminals. Errors are reported on the monitor terminal, but this is often insufficient to detect their cause. Additional information about errors can be found in the data structures of the port expander software. The port expander console process (PECON) provides a set of operator commands that make this information easily accessible, as well as making the internal packet flows visible when problems or questions arise.

A. The PECON Process

The port assignments and routing table for the port expander are maintained as static data structures. Since user requirements may change from time to time, PECON allows operator-entered changes to the original port allocations and routing table, as long as neither the number of ports nor the size of the routing table increases. Some debugging facilities are also provided by PECON: 1) a port can loop (echo) traffic to test the 1822 interfaces, 2) all traffic through the port expander can be traced on the monitor terminal, 3) attached devices can be halted for debugging without losing context, and 4) the port expander can be halted so that the DDT debugger, if present, can be used to check for corrupted code.

Printouts of two types of dynamic data structures are also available through interactive PECON commands. First, there is information on ports and transfers--such as the status of the 1822 interface, the number of packets sent and received from a port, and whether the IMP or NCP host is up. Second, there is specific information on the port expander itself--such as the length of queues, the amount of memory that has been used or is still free, and the port expander ID. The extent to which commands are useful for a specific task or question will depend on the user's familiarity with the port expander data structures. These data structures will be explained, along with other details of port expander functions, in Section VI.

In summary, the facilities provided by the PECON command set include

- Display of port expander status.
- Display of memory utilization.
- Facilitation of hardware/software debugging for the attached hosts, the IMP, and the port expander.
- Alteration of routing and port tables.

B. Command Summary

PECON supports two levels of console commands. The top-level commands provide status, debugging tools, and access to the second level. The second-level commands make changes in the static data structures.

In the following summary, <delim> is a delimiting character such as a space or comma. Carriage return is indicated as <cr>. To execute a command, only enough characters need be typed in to make it unambiguous, as indicated by the underlining of characters in each command. In the on-line help strings, accessed by entering a question mark (?) or the word "help," the minimum characters needed in a command are parenthesized; this on-line listing of the command set is quite convenient to use.

The top-level commands are typed in response to a '-*' prompt. Commands at the left margin in the following descriptions are at the top level:

break Halts the port expander--If the DDT debugger is resident, the process will drop into DDT; otherwise it will drop into ODT. The <esc>P command to DDT, or P to ODT, will cause the port expander process to resume.

cncp Changes the NCP port designation--Either the port to which the NCP host should be attached can be changed, or no NCP host can be declared for the current configuration. The sequence for this command may be as follows:

```
-*cn<cr>
New NCP port (or none)? 2<cr>
-*
```

or shortened to

```
-*cn 2<cr>
-*
```

cportstatus Declares a port either on-line or off-line--The following information is then requested:

. Port number to be changed

. New port status (or quit)

The command may be executed as follows:

```
-*cp<cr>
Port to be changed:  1<cr>
(On)line, (Off)line, or (Q)uit? off<cr>
Port 1 is off line
-*
```

or the command may look like this:

```
-*cp 1 off<cr>
Port 1 is off line
-*
```

or any combination of the above.

croutes Changes any one or all of the route table entries--Both the specific field of the route table and its mask may be changed. One response must be made to the following prompt:

Route item number to be changed:

Route items have implied numbers, counting sequentially from zero at the top of the screen display. The possible responses to the above prompt are

```
quit
all
<n>, indicating a route item
```

One or more responses may be given to the subsequent prompt, "&". They are

```
quit
host<delim><oct>-changes host number
hmask<delim><oct>-changes host mask
imp<delim><oct>-changes IMP number
imask<delim><oct>-changes IMP mask
lhost<delim><oct>-changes logical-host number
lmask<delim><oct>-changes logical-host mask
net<delim><oct>- changes host number
nmask<delim><oct>-changes net mask
port<delim><oct>-changes destination port number
```

ctime

Sets a time interval, in seconds, for accumulating data on a port. For example, the time is initialized to one second, so that every second the number of packets sent and received in the preceding second is averaged and kept. The command sequence is

```
-*ct<cr>
Time between polling of send and receive counts:  2<cr>
-*
```

or shortened to

```
-*ct 2<cr>
```

debug

Turns on packet printing (tracing) at the monitor terminal. WARNING: The cost in port expander efficiency is high, so the function should not be left on any longer than necessary.

help

Lists the PECON commands, with brief explanations. The minimum keyboard entry for the command is shown in parentheses. A question mark (?) is equivalent to the "help" command.

loopback

Sets any one port on the port expander to loopback (echo) mode. In this mode, all packets received on the looping port will be sent back to that port regardless of packet content.

lportstatus Provides details on the status of all PE ports, i.e., whether a port is up, on line, down, polling, etc. The IMP is always port 0.

lptype Lists the type of device to be attached to each port of the port expander. For example, an NCP port expects an NCP host to be attached to it.

lqstatus Lists the lengths of the send queues for all ports, the RFNM queue, and the connection queue. See Section VI for details.

lroutes Lists the contents of the route table, minus the masks. The order of listing is the net number, host number, logical-host number, IMP number. A zero entry means the field is ignored. Each port can have a separate entry for each net number.

ltime Reports (1) the average number of packets sent and received over the time set with the "ctime" command, as well as (2) the total of packets sent and received since the port expander was last started. All statistics are expressed in decimal form.

memstat Reports information about the port expander's global memory usage. The statistics are expressed in decimal numbers of bytes.

nodebug Turns packet printing off. See the "debug" command.

noloop Restores all ports on the port expander to normal function. See the "loopback" command.

nottardy Turns off tardy checking for a port (the "pcomingup" field in the port structure) so that transmission of a packet can be indefinitely long. See the "tardy" command. WARNING: The port expander has limited buffering abilities, so if an attached host is halted and packets keep coming in for the halted host, at most 15 will be kept. If the 16th packet arrives, the oldest packet in the queue (i.e., the first packet) will be discarded. This process will continue for as long as packets come in and the queue is full.

The command sequence is

```
-*not<cr>
Do not check for tardy condition on port:  1<cr>
-*
```

or shortened to

```
-*not 1<cr>
-*
```

tardy — Returns a port to the normal state of checking for a tardy condition. See the "nottardy" command.

The command sequence may be

```
-*t<cr>
Return to checking for tardy on port:  1<cr>
-*
```

or shortened to

```
-*t 1<cr>
-*
```

whoami — Gives the current host/IMP identification and reports the state of the NCP host and IMP.

C. Monitor Messages

There are two types of messages displayed on the monitor terminal: (1) trace messages, which display a message string along with portions of the ARPANET leader contents, and (2) diagnostic messages, which display a port number, if appropriate, along with a message string. Both types are intermixed in the alphabetized list of messages below. The messages have the following formats:

```
Trace messages:  "<message>-- Host/IMP:<oct>/<oct> Link:<oct>
     Type:<dec> Port:<oct> Op:<oct>"

Diagnostic messages:  "Port <oct> - <message>"
```

The "host", "IMP", "link", and "type" values are taken from the ARPANET leader of the packet. The "port" value is the port number the packet was sent to if the packet was routed by the port expander; if the packet was not routed by the port expander, "port" is the port number that the packet came from. "Op" is the signal opcode value associated with the packet. The symbols within the "<>" are variable text: <message> is the message string, <oct> an octal number, and <dec> a decimal number. Trace messages are printed when the "debug" command is entered; they are not printed after the "nodebug" command has been entered, except that error and certain status messages can never be turned off.

The complete set of messages is listed below. Trace and diagnostic messages are identified as such at the end of their descriptions.

"Bad msg count in Connq" - Indicates a corrupted outstanding message count in the CONNQ structure. This indicates imminent software disaster. (diagnostic)

"Bad signal opcode in main" - This indicates that a signal opcode value was received that is not within the correct range for the main loop of the port expander process. Hardware and/or software corruption should be strongly suspected. (diagnostic)

"blocked" - The port is blocked for flow control. This message will appear only if the "debug" command has been entered. (diagnostic)

"Confusion in output queue" - Inconsistencies were found while the output queue was being flushed to the tardy host. This message appears after a "host tardy ..." message and indicates a corrupted send queue. (diagnostic)

"Dead host element expired" - The long-awaited dead host status message promised by the destination dead message did not arrive. This message is generally not useful. (diagnostic)

"Error in leader" - An error in the ARPANET leader packet was received from the IMP. This message occurs most often during initialization on the IMP port. (diagnostic)

"Error in leader received" - An error in the ARPANET leader packet has been received from the indicated host port. This message also occurs with hardware loopback. (diagnostic)

"Error on a port" - A buffer was received from the operating system's I/O subsystem with the error bit set in the IORB data structure. This message routinely occurs whenever the indicated port initializes or the port goes down. (diagnostic)

"Error in transmission" - A port error occurred while the message was being transmitted and can also occur when a port goes down. The source of the message, if it is localized in the PE, is notified with an incomplete-transmission packet. (diagnostic)

"Host tardy, port taken down" - As the indicated port has taken more than 15 to 20 seconds to receive the message, it is declared down and tardy. (diagnostic)

"Illegal message type" - A type-3 message was received from the IMP. Type-3 messages are legal only with 32-bit leaders and should never appear. (diagnostic)

"No port found for RFNM" - The scan of the RFNMQ could not find an entry to match the received RFNM. This error indicates either corrupted tables or duplicate RFNMs. This is a message that should never be generated. (diagnostic)

"No route for message" - There is no route in the routing table for this message from the IMP or the port to which the message should be sent is down (the NCP port is either down or disabled). Such messages are flushed with no network notification. (trace)

"NOP from host" - A host-generated NOP message was received. (trace)

"No route for this host packet" - There was no route available or the message should go to the IMP and the host was notified with a destination unreachable message. (trace)

"NOP from Imp" - Indicates that a NOP was received from the IMP; the address displayed is that of the PE. This is a handy message if the various IMP cables get confused. (trace)

"Output queue overflow" - The most current packet that arrived has exceeded the size limit of the send queue and has forced the PE to flush the first message in the queue. If the message was generated by one of the attached hosts, that host is notified with an incomplete-transmission message. If the message was from the IMP, it is quietly discarded. (diagnostic)

"Received old format msg" - The flag field of the ARPANET leader did not indicate 96-bit format. Non-96-bit messages are flushed by the PE. (diagnostic)

"Rfnm overdue" - The 72-second timeout has expired and the PE, having become impatient waiting for a RFNM from the network, has generated an incomplete-transmission message and forwarded it to the data message source. The PE has assumed that the RFNM got lost in the net. This diagnostic can occur when the PE-IMP connection goes down and the IMP does not send an Interface Reset (a very rare occurrence). The more likely reason, especially if it occurs frequently, is that the IMP has been delayed by net topology or load and the timeout is too short.

"Route msg" - Reports the transfer of a packet. It is under the control of the software DEBUG switch, which is set on with the "debug" command. It traces all packets as they are sent. (trace)

"Salvage space for a connq entry" - The PE was attempting to create a new CONNQ entry, but failed. This is a panic situation that causes the PE to search for space. It occasionally finds some, but it is usually so saturated that it is forced to restart. (diagnostic)

"SRI port expander V4.1.0, Jan. 1980" - This message is printed when the port expander process is started. The operating system (MOS) has successfully loaded and started processes, or the system had to restart when this herald appeared. It is neither a trace nor a diagnostic message.

"Unblock" - The PE has unblocked a previously blocked port. This message will appear only if the software DEBUG switch is turned on with the "debug" command. (diagnostic)

"Unknown message type" - The message is out of the message-type range expected from the IMP. These message types are part of the unimplemented nonblocking interface. (diagnostic)

"Unknown msg type" - The message being processed from the indicated host port is not a legal type. This message will always occur when traffic flows through a host port that is hardware-looped with wire jumpers. The reflected RFNM messages cause the PE to emit this message and flush the offending message. (diagnostic)

VI PORT EXPANDER SOFTWARE

The port expander process software is written in Bliss-11 and runs under the MOS operating system. All routines are documented in the program source; each routine is prefaced by text that contains a condensed version of the algorithm used and the interactions with other routines.

The program source text is called <PKT-PE>IMPMUX.B11. The discussion that follows is written for users experienced in Bliss-11 and ARPANET protocols.

A. Port Expander Data Structures

All PE data items except local temporaries or a few global cells are contained in either the two static structures, $PORT and $ROUTE (defined in the file ROUTES.B11), or in the two dynamic structures, RFNMQ and CONNQ (defined in IMPMUX.REQ). The reader is referred to the source files for detailed descriptions of the data fields and their use.

1. Static Data Structures

There are two static tables that define the system configuration: $PORT is the interface with the 1822 devices, while $ROUTE is the packet-routing table.

The port table ($PORT) has an entry for each possible network device (host computer or gateway) configured into the port expander software. Each port table entry contains the following 15 fields:

Length (bits)	Field	Description
16	input	MOS device control table input offset
16	output	MOS device control table output offset
8	pstatus	port status (the bits are detailed below)
8	ptype	port type: IMP (6), NCP (1), or INTERNET (2)

8	pcomingup	see the "not tardy" command in Section V
8	padding	the number of 16-bit padding words needed by the host for every packet
16	up	the number of times the host came up
16	down	the number of times the host went down
16	blocked	the number of packets from the host currently being processed by the PE
16	recvd	the number of packets received by the host
16	sent	the number of packets sent to the host
16	recvderr	the number of packets received in error
16	senterr	the number of packets sent in error
16	outqhead	the head address of the send queue for this port
16	outqtail	the tail address of the send queue for this port
8	outqcnt	the count of packets outstanding and in the send queue
8	outqtime	a timer for send completion, which counts down from 15 seconds

The port status (pstatus) bits mentioned above are:

Bit Number	Bit
0	ptinit
1	pttardy
2	pthopen
3	ptiopen
4	ptblocked
5	ptreceive
6	pttest
7	ptoffline

Each port is controlled by the off-line bit ("ptoffline") of its status. If this bit is set, the software will completely ignore the port; setting the bit on eliminates unnecessary processing time. When a port is functioning normally, the status byte will be 0 or it will have "ptreceive" set, which means that a new receive needs to be posted. If "ptopen" (dist-relay open) and "pttest" are set, the port is down and the PE is polling to test for port up. If "pttardy" or "pttardy" and "ptopen" (local relay open) are set, the last send failed to complete; the port will be down until an initializing send is completed and/or a good packet is received. "Ptblocked" indicates that the port is blocked until a RFNM is received on an outstanding connection--that is, until the number of packets outstanding on a connection is greater than three (at present) or the number of packets outstanding from any host is greater than six (at present). The table is sensitive to changes in the DCT tables of MOS. Consequently, implementors who use a nonstandard MOS operating system are warned to ensure that this table will track all MOS changes, because entries in the tables are initialized to constants set by the standard MOS configuration files.

The routing table ($ROUTE) is a sequentially searched data base containing mask and match words that correspond with internet protocol (IP4) destination address fields, as well as with a port number field used for the result of a successful match. The ordering of the entries is important, because the search is sequential and stops at the first match. See below for the routing algorithm.

Two values control the configuration: IMPPRT (normally set to 0) and the variable NCPPRT determine which port is to be used for the IMP and NCP host, respectively. NCPPRT has an additional function in that a negative value, (i.e., -1) indicates the absence of an NCP port. This affects the routing algorithm only.

2. Dynamic Data Structures

The dynamic structures contain no configuration-dependent information and are therefore of no interest in a simple installation. Only implementers who plan to integrate the port expander with other software need to understand the dynamic data structures before they can introduce modifications.

All data necessary for managing packet traffic are contained in one of two dynamic structures. Each entry is allocated on demand and returned to free storage when it is no longer needed. This enables the port expander to respond to shifting traffic patterns with a minimum storage requirement. Both structures are implemented as single-link FIFO lists with head and tail pointers.

The RFNM queue (RFNMQ) structure is used to track each packet that is sent. An entry is created and inserted in strict FIFO order whenever a data packet is transmitted from a host to either the IMP or another host. The entry is deleted whenever a RFNM is received or generated. The RFNMQ information includes the host-IMP-link fields and the port number of the packet source. The latter is necessary for routing the RFNM back to the source of the data packet. See below for the algorithms used to process RFNMs.

The CONNection queue (CONNQ) structure is used for control of a host-to-host connection. An entry is created for each destination host. Any traffic that cannot be forwarded directly to the destination host is queued onto the CONNQ. Connection queue entries contain the host-IMP ID, a count of outstanding messages, the age of the connection, and a queue of packets that cannot yet be sent. Up to three outstanding packets for each destination host from any other host on the PE are queued. The queue is used to block a source host and to store the last (fourth) packet from each source host.

B. Port Expander Program Flows

The PE program flows include initialization, responses to data packet reception, responses to data transmission completion, and periodic housekeeping tasks scheduled at regular intervals.

1. Initialization

Initialization is done by the INIT routine. It resets the traffic counters and sets the port status bits ('pstatus') in $PORT to declare all the ports down. The port status bits are later used by the service routine to complete initialization. The INITPORT routine starts the protocol dialogue with the connected hosts and the IMP. As blocked output to a host can severely deplete buffer space, port initialization is performed in stages to ensure that the host will be ready to accept packets.

2. Packet Reception

All packets are first processed by the MSGIN routine to do preliminary packet filtering. Its main functions are to control the state of the port, call the HOSTIN or IMPIN routine, and (if there are no errors) report a receive on the port. Port errors are trapped and flagged for processing in one-second intervals by the SERVICE routine.

MSGIN passes a successfully received packet to a lower-level routine for port-type-specific processing and assumes that the lower-level routine will dispose of the packet buffer. It will then resume communication with the port by calling the RECEIVE routine, except when the port has been blocked because the number of packets to one destination is greater than three (at present). If there are no buffers available, the RECEIVE routine sets "ptreceive" in the port structure, which signals the SERVICE routine to retry later. In the port expander, only one receive will be posted on a port at a time.

a. IMP-to-Host

Packets that arrive at the PE from its IMP are processed by the IMPIN routine. The packets fall into two general categories: data packets that must be forwarded, and IMP-Host packets that control the state of the ARPANET protocol. All packets are either forwarded to the appropriate host port or, in the case of a dead host, broadcast to all active host ports. Figure 6 illustrates this process.

b. Host-to-Host

Data packets that originate from PE host ports are processed by the HOSTIN routine, as illustrated in Figure 7.

Each data packet is accounted for in a connection queue (CONNQ) entry. It is either sent and counted or, if the port is to be blocked, counted and then queued under the connection entry. This queueing strategy is necessary to prevent the IMP from having to block the port expander because more packets than the maximum allowed number

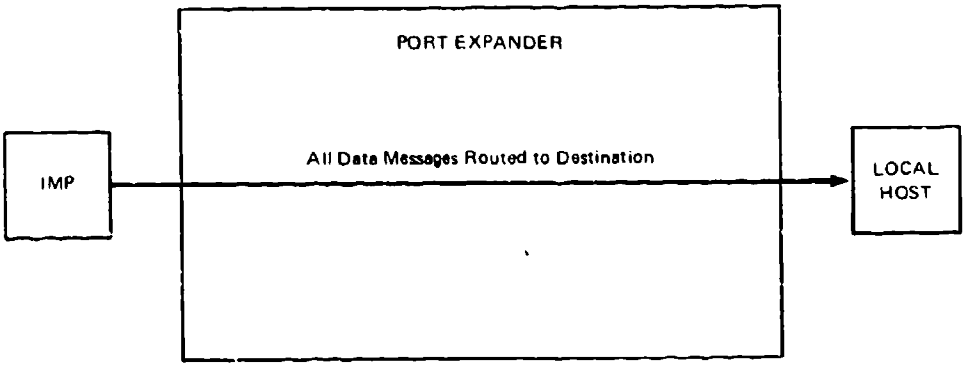


FIGURE 6 IMP-TO-HOST FLOW

(typically eight) are in transit to the same destination host. The MAXOUTSTANDING numbers are changeable; their current value (three) can be found in the IMPMUX.B11 source file.

An entry is made in the RFNM queue (RFNMQ) whenever the SEND routine moves a data packet to the MOS I/O subsystem. There is one entry in RFNMQ for every packet that has been sent in anticipation of an acknowledgment by the network that the destination host has received the packet. Local store-and-forward routing is done by emulating the IMP (and network) and generating a RFNM internally after the data packet has been sent to the destination host.

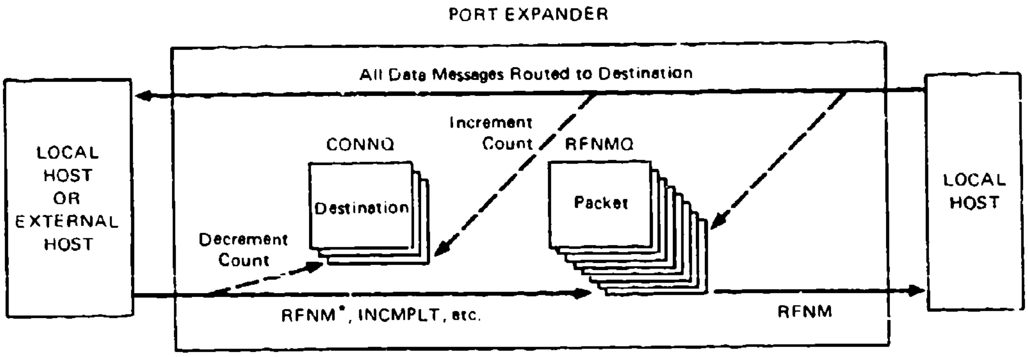
Irregular messages from the host are processed within HOSTIN and, in general, not forwarded to the IMP. The exception to this is the Host-Going-Down message from the NCP port that is delayed until all outstanding traffic has been sent to the IMP.

3. Packet Transmission

Three events can occur when a packet is transmitted by the PE. The packet transmission can be completed normally, be in error, or be aborted by a tardy timeout. Packets can also be aborted prior to the transmission if the send queue to a port becomes saturated.

The output completion signal from the MOS I/O subsystem is processed by the PORTOUT routine. First, the packet is tested for successful transmission. A reply is created and returned to the source of a data packet when IMP emulation is required of the PE for local host-to-host transfers. The packet buffer is then returned to the buffer pool. A RFNM is created if the transmission was successful; an incomplete-transmission packet is created if it was not. The new message is processed by RFNMIN just as if it had been received from the IMP.

A port can be declared tardy by the SERVICE routine if a packet transmission takes longer than 15 seconds. It will then have to go through the initialization sequence before data transmission can resume. See the following section on periodic processing.



* The RFNM is generated by the IMP if the message destination is an external host. If the destination is another local host, the port expander fakes a RFNM to the source host.

FIGURE 7 HOST-TO-HOST FLOW

The first packet waiting for transmission in the send queue to a port can be flushed if a new packet arrives for the port and the queue is already full. An incomplete-transmission packet is sent back to the packet source to inform it of the failure. This is reported to the operator with an "Output Queue Overflow" message on the monitor terminal.

4. Periodic Processing

System consistency checking, dynamic storage recovery, and port initialization are done in sequence by the SERVICE routine. The SERVICE routine is called via a timer signal from MOS; upon completion of its tasks it resets the timer to signal again.

The JOLT routine handles queue servicing. Entries in the RFNMQ are aged to detect stale (72-second) RFNMs that may have been lost in the network. An incomplete-transmission packet is generated and the RFNMQ entry is removed whenever the 72 seconds elapses. All packets received from the host ports are accounted for by this strategy and will get a negative acknowledgment even if the IMP dies or the RFNM gets lost. This algorithm is also used to remove stale destination-dead entries in the dead host queue that were kept in anticipation of dead host status messages. These entries are simply reported to the operator and flushed; no response is sent to the waiting host ports.

The CONNQ entries are also aged for 15 seconds. An aging mechanism is used on CONNQ entries to allow the entries to remain in abeyance in anticipation of more traffic on a slow connection, as well as to prevent unnecessary storage allocation-deallocation thrashing. CONNQ entries are removed if they are inactive and/or the free pool is exhausted.

The status of all polling ports is examined every time the SERVICE routine is called. The "ptreceive" bit of "pstatus" is examined and the RECEIVE called whenever MSGIN or SERVICE (on a previous call) was unsuccessful in getting a packet buffer. If the PE is thrashing and no buffer space is available, there is no upper bound on the time a port could be left idle; SERVICE will attempt to post the receive once every second. The initialization flag "ptinit" is also examined and the necessary initialization done. Port output is timed out and the port declared tardy ("pttardy" status bit) if a transmission takes more than 15 to 20 seconds to be completed. Every waiting packet will be completed in error and an incomplete-transmission packet sent back for each packet to its source.

5. Routing

All routing decisions are made by the ROUTER routine. The algorithm is documented in the program source. The value returned from ROUTER is either a destination port or the value NOROUTE, which indicates that the packet will be reported and discarded.

The use of an NCP port is significant only for routing. It has two effects: any traffic that cannot be routed to another PE host is sent to the NCP port; and forwarding of internet traffic to the IMP is prohibited if the NCP port is down. This latter effect is necessary in order to make remote NCP protocol hosts work properly, since NCP requires information from the network about other hosts. The NCP port special effects can be disabled by setting the NCPPRT variable in the ROUTES data structure to a negative number (e.g., -1).

Internet traffic is routed according to information in the destination address field of the internet header (bits 160 through 191). The routing table ($ROUTE) is searched sequentially and, if a match is found and the port is both up and not the IMP port, then the port value is returned. This last condition is an extra precaution that prevents routing circles (e.g., The address field is such that the PE routes the packet to the IMP, which routes it back to the PE to be routed back to the IMP...).

This algorithm can be used to filter packets to different ports, depending on a subset of the internet destination address field.

If the routing table entries are ordered so that specific match conditions are encountered before the more general match conditio s certain addresses may be then be filtered out of the general traffic stream. For example, this capability could be used where there are two gateways attached to the PE and the adjoining nets are also interconnected elsewhere by means of gateways.

6. RFNM Processing

RFNM processing is integral to the multiplexing function of the port expander.

All ARPANET packets are processed by the subnet through the positive/negative acknowledgment scheme of RFNM, incomplete-transmission, and destination-dead packets. The subnet must acknowledge receipt of packets, and in fact, will go into an error state if they are lost. The PE processes them all with the same degree of importance and will go into just as acute an error state if they get lost in the PE. The PE protects itself from these errors by aging the RFNMQ associated with the transmitted data packets and by generating packets to notify the source port of the error. See the foregoing section on periodic processing.

All data packets sent by a host port, regardless of destination, are tracked through an entry in the RFNMQ. These entries are created in the SEND routine whenever it outputs a data packet that has been received from a host port. They are inserted in FIFO order.

The RFNMQ entries are removed by the RFNMIN routine. The RFNMQ is searched sequentially from the front until the first match is found, because RFNMs usually return from the destination IMP in FIFO order; however, RFNMs from different destinations get mixed because of the different hop distances to different destinations. The routine also uses the connection entries (CONNQ) to decrement the outstanding message counts, send the next packet that has been waiting in the queue, and free blocked ports. The RFMNIN routine processes all RFNM, incomplete-transmission, and destination-dead packets identically, except that queuing for destination-dead packets is done in anticipation of receiving dead-host-status packets.

The RFNMIN routine is called by IMPIN to process these messages from the network, by PORTOUT to do IMP emulation of fake RFNMs (local host-to-host) generated by PORTOUT, and by JOLT to emulate the network for RFNMs that are believed lost.

7. Trace Printing

For system monitoring, packets in error or with no route available to them are logged on the terminal (if any) attached to the monitor port, as described at the end of Section V. The "debug" command at the console causes all packets sent to be logged onto the monitor terminal. A monitor printer is not required for normal PE operation and can be disconnected from the system at any time by uncabling the terminal from the PE. CAUTION: Do not turn off either the monitor terminal or console terminal until it has been uncabled from the PE.

VII USER SUPPORT AND MAINTENANCE

SRI will provide limited assistance in planning deployment of the port expander facilities by responding to unresolved questions. Users are responsible for the fabrication of cables needed to connect the port expander to the IMP and to the attached hosts, but the appropriate bulkhead connectors are provided with the port expander itself. Users are also responsible for providing an appropriate mounting location for the port expander hardware, as described in Section IV.

Once the facilities have been prepared, SRI personnel will provide on-site installation and checkout of the port expander hardware to ensure its proper operation in conjunction with the user's equipment.

The exact details of equipment maintenance are still being developed, but our experience to date with the hardware indicates that it is very reliable in operation.

Telephone assistance with problems will be provided after on-site installation.

A. The 1822 Interface LEDs

The LEDs on the 1822 cards are (from left to right in the standard DEC box and from top to bottom in the ruggedized box):

xmit-dma rcv-dma local-relay dist-relay bus-back

Although this is not the standard way of describing the relays, it appears to us to be clearer and more easily comprehensible.

For a port with nothing connected to it, local-relay should be on and all other LEDs should be off.

For a port with an active host connected to it, local-relay and dist-relay should be on, and xmit-dma and rcv-dma should be flashing.

The rcv-dma will be on most of the time, but should occasionally blink off when the host is sending.

For a port with a TARDY host (host has not dropped its relay, but is down) dist-relay, local-relay, rec-dma, and xmit-dma should be on continuously.

The bus-back should never be on.

If the port has been declared off-line, (the procedure for doing this is described in Section V) no lights will be on; this means that the DMA controller board is ignoring that 1822 board.

If the lights differ from the pattern described here, there is something wrong that requires investigation and ultimate correction.

Appendix A

GLOSSARY

Appendix A

GLOSSARY

Blocked--A host computer is blocked when it is prevented by its IMP or PE from sending messages. This typically occurs when a destination host is not responding fast enough to packets previously sent (as a rule, more than three packets outstanding for a destination). The PE tracks outstanding packets and will block its own hosts, if necessary, before the IMP blocks the PE.

CONNQ--Connection queue. There is one CONNQ in each PE. Its function is to count outstanding packets to each destination host from each local host, using that total as a basis for determining when to block a source host. Fields in the CONNQ include (1) link to next entry, (2) destination IMP ID, (3) destination host ID, (4) message count, (5) age, (6) head pointer, and (7) tail pointer.

Dead Host Queue--There is one dead-host queue in each PE. It stores dead-host messages from the IMP for a period of about 72 seconds, after which the messages are discarded.

DDT--A symbolic debugger program.

DCT--Device control tables of the MOS operating system.

Gateway--An internetwork processor for transferring messages between two networks.

Host--Any computer or gateway attached to a network.

IMP--Interface message processor.

Internet protocol--One of the control protocols established by DARPA for transmission of messages between two networks.

Leader--An ARPANET message header.

Link number--The low byte of the ARPANET host-to-IMP leader's message ID field (bits 65 to 72).

Local host--A host directly attached to the PE.

Monitor terminal--The output-only terminal used to list trace and diagnostic messages.

MOS--Micro operating system developed by SRI.

NCP--Network Control Protocol (ARPANET-specific).

NOP--No operation.

Output queue--Send queue.

Outstanding packet--A packet sent to a destination with no RFNM in response.

PECON--The port expander console process, which allows the operator to investigate and alter software tables through interactive commands.

RFNM--Ready-for-next-message packet. These are generated by the IMP (for external-destination messages) or by the PE (for local-destination messages).

RFNMQ--The RFNM queue in the PE. There is one RFNMQ in each PE. Its function is to store space for a RFNM pending its receipt from the destination. Fields in the RFNMQ include (1) link to next entry, (2) destination IMP ID, (3) destination host ID, (4) packet destination link number, (5) age, (6) source port.

Send queue--The output queue. There is one send queue for each port in the PE. It queues incoming packets (up to 15 per port) and serves as a buffer for forwarding.

Subtype number--Bits 77 through 80 of the ARPANET message leader.

TCP--Transmission control protocol developed by DARPA as a successor to NCP. TCP can be used for internetwork traffic, whereas NCP is confined to the ARPANET.

TIU--Terminal interface unit. See the TIU notebook.

Trace--A message printed on the monitor (output only) terminal that traces packet movements.

Word--Sixteen bits.

1822--ARPANET IMP-host interface specification developed by BBN. The PE uses 1822 distant host interfaces exclusively.

Appendix B

RS-232C PINOUT

Appendix B

RS-232C PINOUT

EIA INTERNAL CABLE

Berg 2x5 Pin Connector		EIA DB25 Connector
1	NC	
2,5,9	ground	1,7
3	transmit data +	3
4	transmit data -	1,7
6	key	
7	receive data -	1,7
8	receive data +	2
10	nc	

Appendix C

1822 CABLE

Appendix C

1822 CABLE

Berg 2x13		AMPHENOL
Pin Connector		MS24264R18831SN
1	"last IMP bit" 1	21
2	"last IMP bit" 0	22
3	IMP data 1	23
4	IMP data 0	24
5	"there's your IMP bit" 1	19
6	"there's your IMP bit" 0	20
7	"ready for next IMP bit" 1	7
8	"ready for next IMP bit" 0	8
9	NC	
10	NC	
11	host ready test	13
12	host master ready	14
13	IMP ready test	11
14	IMP master ready	12
15	NC	
16	NC	
17	"ready for next host bit" 1	17
18	"ready for next host bit" 0	18
19	"there's your host bit" 1	5
20	"there's your host bit" 0	6
21	"last host bit" 1	1
22	"last host bit" 0	2
23	"host data" 1	3
24	"host data" 0	4
25	NC	
26	ground	31